CONGRUENT SOFTWARE, INC.
3998 Whittle Avenue
Oakland, CA 94602
(510) 531-5472
(510) 531-2942 FAX

FROM: Peter Johansson

TO: T10 SBP-2 *ad hoc* Working Group

DATE: September 18, 1997

RE: Bi-directional ORB

Several of the 1394 printer working groups have expressed an interest in an ORB that is more suited to the nature of data flow between initiators and printers. The suggested bi-directional structure may be useful for other classes of device, such as scanners or cameras.

In the proposal below, note two things:

- The ORB uses the same *rq_fmt* value as existing ORB's. The assumption is that a device is implemented to use normal or bi-directional command block ORB's but not both. The type of ORB is determined by context; and
- The input and output data buffers reside at the same node and therefore share the same speed, maximum payload and page boundary characteristics.

5.1.2.2 Bi-directional command block ORB

The format of the bi-directional command block ORB is illustrated by the figure below.

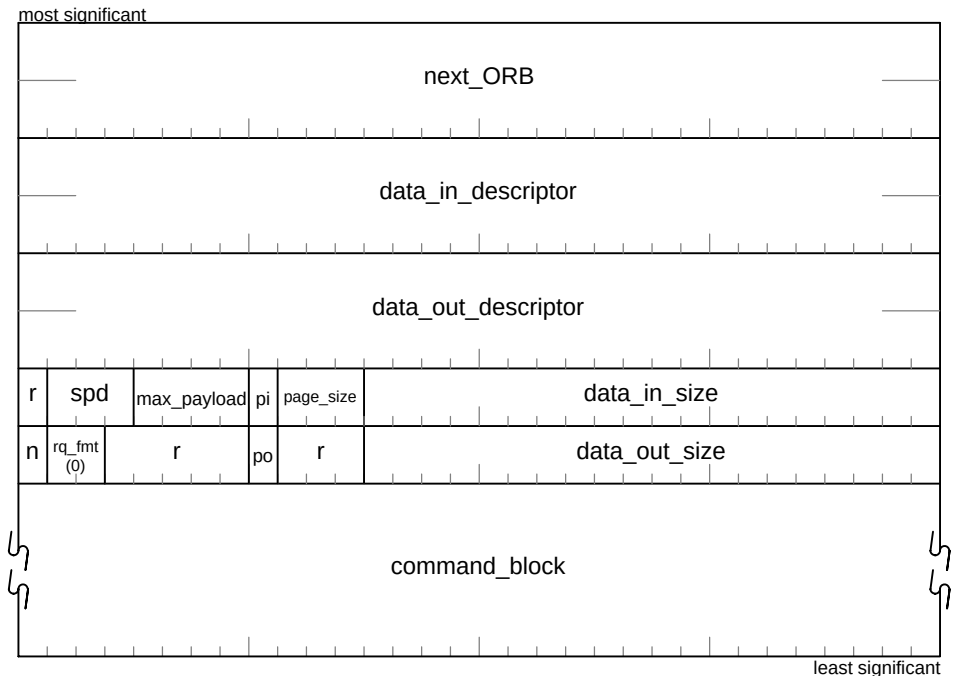

Figure 17 – Bi-directional command block ORB

The *next_ORB* field shall specify a null pointer or the address of a dummy ORB or a normal command block ORB and shall conform to the address pointer format illustrated by Figure 12.

The *data_in_descriptor* field shall specify either the address of the input data buffer or the address of a page table that describes the memory segments that make up the input data buffer, dependant upon the value of the *page_table_in_present* bit. The format of the *data_in_descriptor* field, when it directly addresses a data buffer, shall be a 64-bit Serial Bus address or, when it addresses a page table, shall be as specified by Figure 11. When *data_in_descriptor* specifies the address of a page table, the format of the page table shall conform to that described in 5.2. The target shall use Serial Bus write transactions to store data into the input buffer.

The *data_out_descriptor* field shall specify either the address of the output data buffer or the address of a page table that describes the memory segments that make up the output data buffer, dependant upon the value of the *page_table_out_present* bit. The format of the *data_in_descriptor* field shall conform to the requirements for the *data_in_descriptor* field. In addition, the most significant 16 bits of both fields shall be equal. The target shall use Serial Bus read transactions to fetch data from the output buffer.

The *spd* field specifies the speed that the target shall use for data transfer transactions addressed to the data buffer(s) or page table(s), as encoded by Table 1.

The maximum data transfer length is specified as $2^{max\_payload + 2}$ bytes, which is the largest data transfer length that may be requested by the target in a single Serial Bus read or write transaction addressed to the data buffer. The *max_payload* field shall specify a maximum data transfer length less than or equal to the length permissible at the data transfer rate specified by *spd*.

The *page_table_in_present* bit (abbreviated as *pi* in the figure above) shall be zero if *data_in_descriptor* directly addresses the data buffer. When *data_in_descriptor* indirectly addresses the data buffer, this bit shall be one.

The *page_size* field shall specify the underlying memory page size for both data buffers . A *page_size* value of zero indicates that the underlying page size is not specified. Otherwise the page size is 2^{page_size+8} bytes.

When *page_table_in_present* or *page_table_in_present* is one, the *page_size* field also specifies the format of the corresponding data structure that describes the input or output data buffer . A *page_size* value of zero implies the unrestricted page table format (also known as a scatter/gather list). Otherwise, a nonzero *page_size* indicates a normalized page table.

If *page_table_in_present* is zero, the *data_in_size* field shall specify the size, in bytes, of the system memory addressed by the *data_in_descriptor* field. Otherwise *data_in_size* shall contain the number of elements in the page table addressed by *data_in_descriptor*.

The *notify* bit and *rq_fm*t field are as previously defined for all ORB formats. The *rq_fmt* field shall be zero.

The *page_table_out_present* bit (abbreviated as *po* in the figure above) shall be zero if *data_out_descriptor* directly addresses the data buffer. When *data_out_descriptor* indirectly addresses the data buffer, this bit shall be one.

If *page_table_out_present* is zero, the *data_out_size* field shall specify the size, in bytes, of the system memory addressed by the *data_out_descriptor* field. Otherwise *data_out_size* shall contain the number of elements in the page table addressed by *data_out_descriptor*.

The *command_block* field contains a command descriptor block not specified by this standard.